SCTE ISBE CABLE-TEC EXPO '16

SEPTEMBER 26-29 PHILADELPHIA

A PRACTICAL GUIDE TO IMPLEMENTING SOFTWARE DEFINED DOCSIS

David S. Early, Ph.D.

Chief Architect and Data Scientist

Applied Broadband

SCTE Society of Cable Telecommunications Engineers

ISBE International Society of Broadband Experts

#CableTecExpo

Essential Knowledge for Cable Professionals™

Why SDN?

A bunch of reasons, here is a key one:

- Inefficiency has been the profitable watchword
  - Raw BW in the ground
  - Service bundling
- The age of inefficiency is ending (is over?)
  - Net Neutrality
  - Unbundling
  - OTT Services
  - Fine grained service differentiation

Services of Interest

- Telephony
  - Specifically the setup and management of QoS associated with calls
- Congestion management
  - Dynamic, rule-based implementation of bandwidth controls/limits in response to congestion or individual user abuse
- Enhanced video services
  - Establishing QoS for video services

The Promise of SDN

- System and network resource elasticity
  - The virtualization of DOCSIS low-level services using contemporary SDN architectures enables greater design flexibility in networks of scale while affording discrete upgrades to capacities at time of need.
- A common platform across network applications
  - Capital, operational, and organizational economies of scale result when using multiple applications that access the same network services and related resources in a common way.
- Service delivery agility
  - As a consequence (benefit) of the above, new subscriber services that benefit from QoS treatment in the access network can quickly be supported.

DOCSIS SDN vs OpenFlow SDN

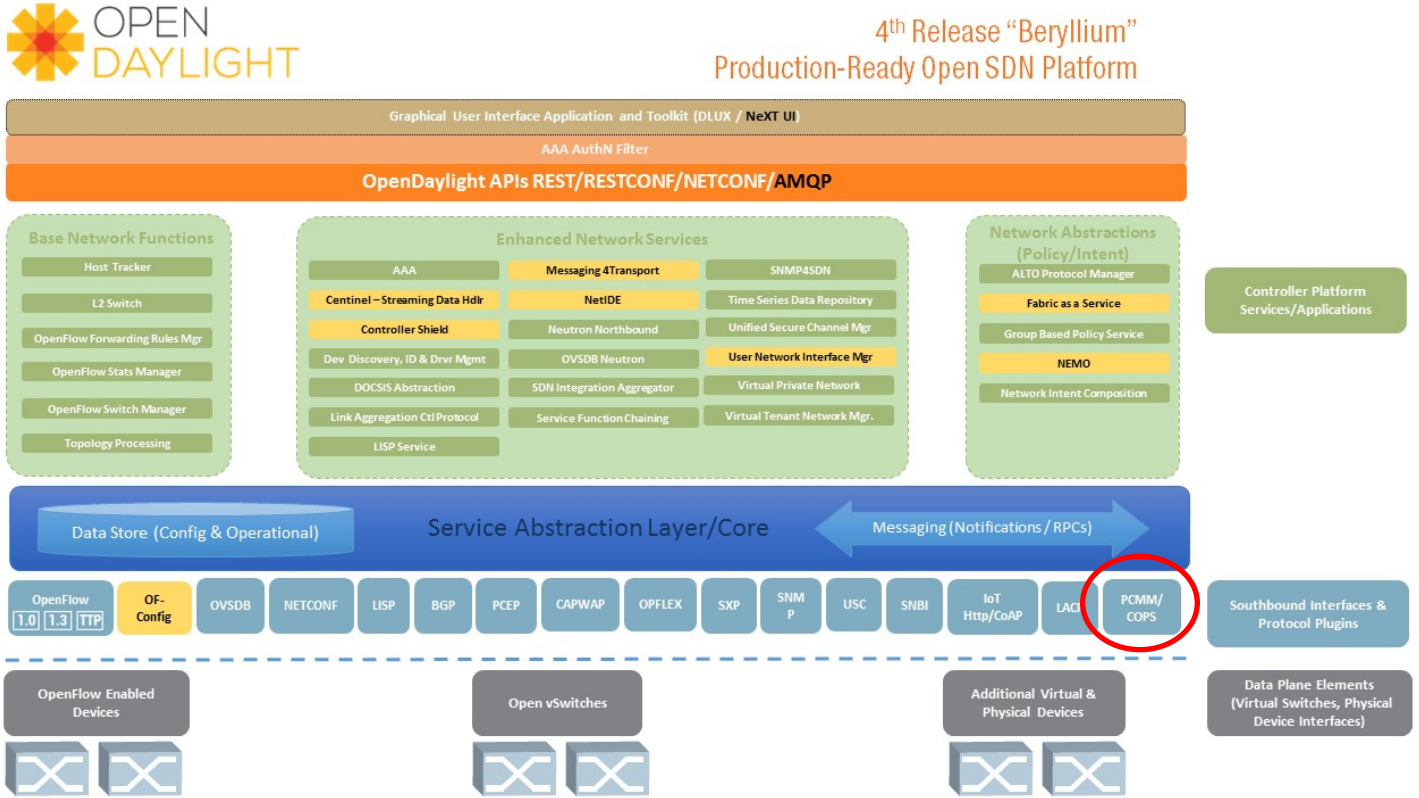
Current efforts have focused on “Traditional” networks

- Topology
  - The CMTS (master) and CM (slave) topology are fundamentally different than port-to-port topology of switched networks (e.g. Ethernet).
- Scale
  - The number of managed devices (entities) in a DOCSIS network is well into the tens of millions and growing.
- Complexity
  - The Information model representing a complete DOCSIS network is much broader than in a traditional IP network

Where do we begin?

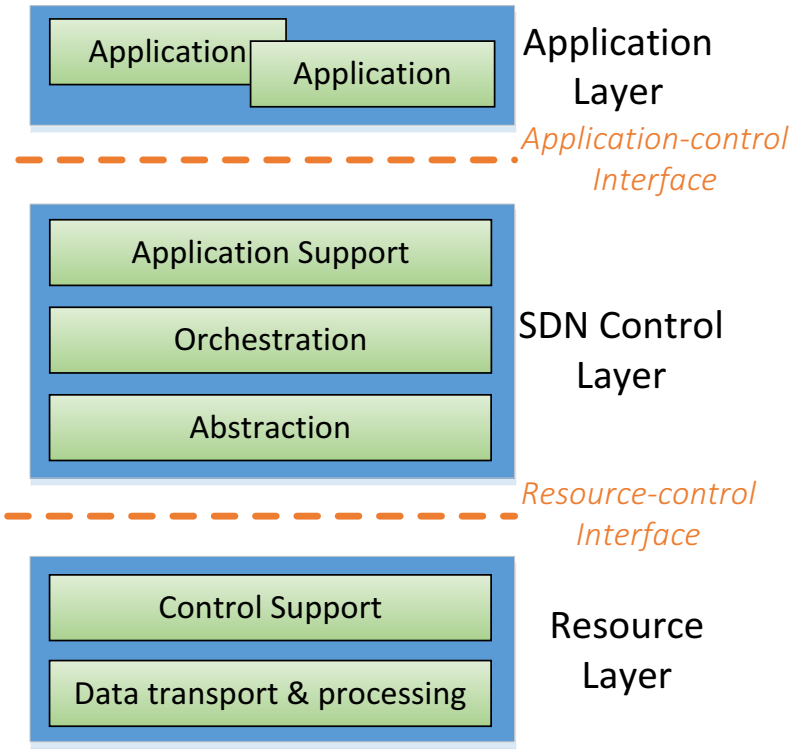
- Leverage the work of others:
  - OpenDaylight (ODL) is an open source SDN controller
  - CableLabs has produced a prototype PCMM module for ODL
  - Active development on ODL
- Leverage Standards/Familiar Territory
  - NETCONF/YANG/REST
- Issues?
  - ODL is a bit OpenFlow centric
  - General maturity, need DOCSIS-oriented features
  - Opensource? Really? (Hint: Yes)

OpenDaylight (ODL): Starting Point

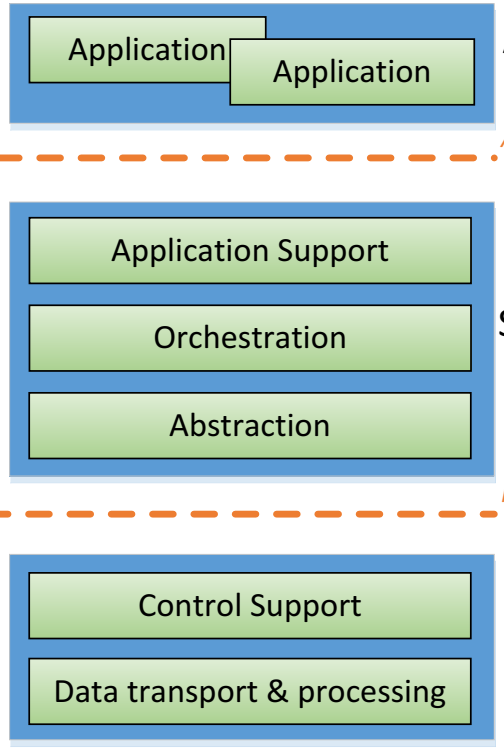


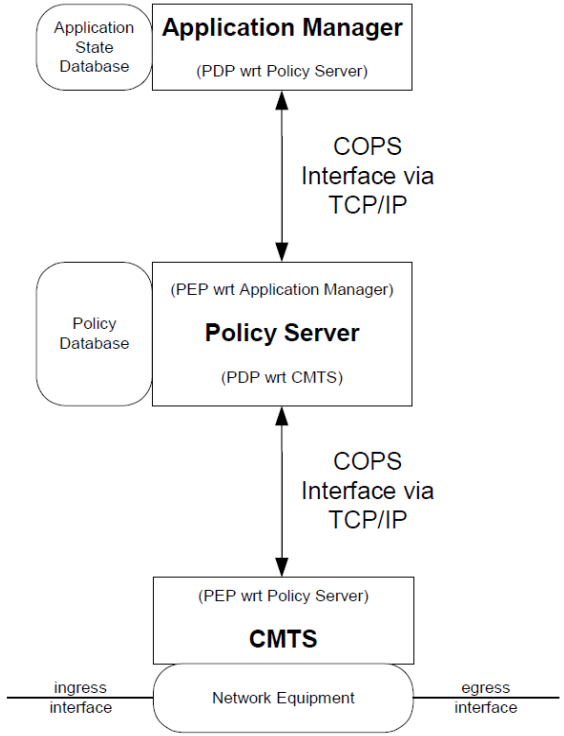
The Formal ITU-T SDN Model

- Application Layer
  - Business Apps
- SDN Control Layer
  - Dynamic control of network resources as instructed by the application layer
- Resource Layer
  - Network devices and element management systems



SDN and PacketCable Multimedia





(From PKT-SP-MM-I06-110629)

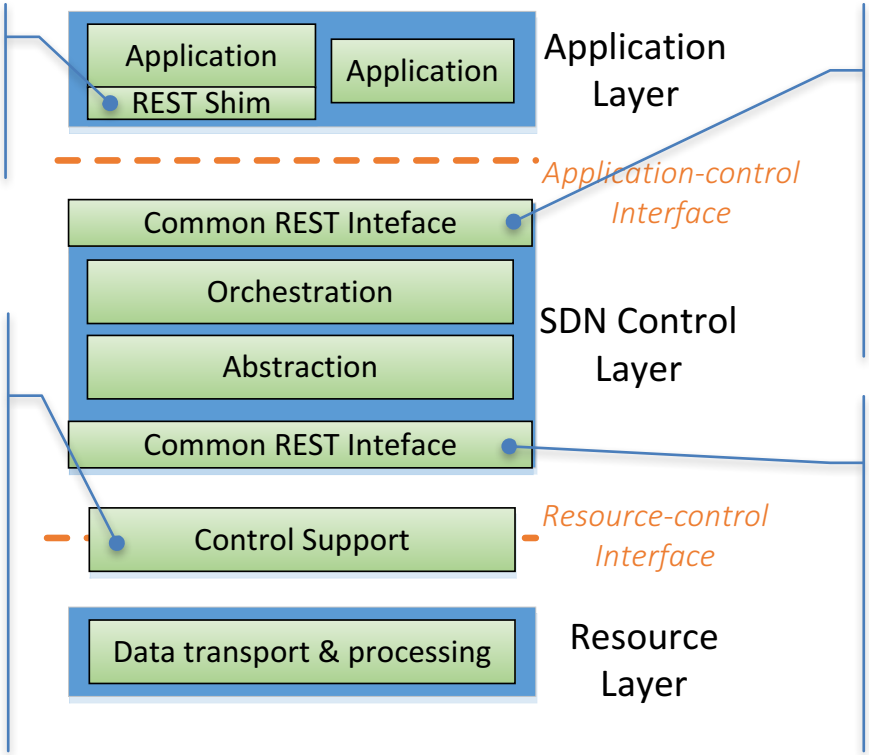
Design Adaption for DOCSIS

- **Common Interface Language**
  - New applications will have a well-defined & extensible interface to the core
- **Limited State in the Core**
  - Stateful information in this architecture is pushed to the edges: Applications will maintain their state, and the edge interface to the network (e.g. in this case, ODL) will maintain state associated with requests minimizing HA/DR requirements on the core
- **Inherent Modularity**
  - With state managed at the edges, core is highly modular.
  - The core's primary function is routing requests to the correct network interface.
  - Incoming request load balancing and HA/DR is straightforward: Send the requests to any available server. If something is down, spin up a new one.
- **Feature Extensibility**

Updated Architecture Model

REST "Shim" allows reuse of existing Apps

ODL (or Equivalent) becomes the Control Support layer which can support PCMM and other protocols. Independent scaling and deployment possible.

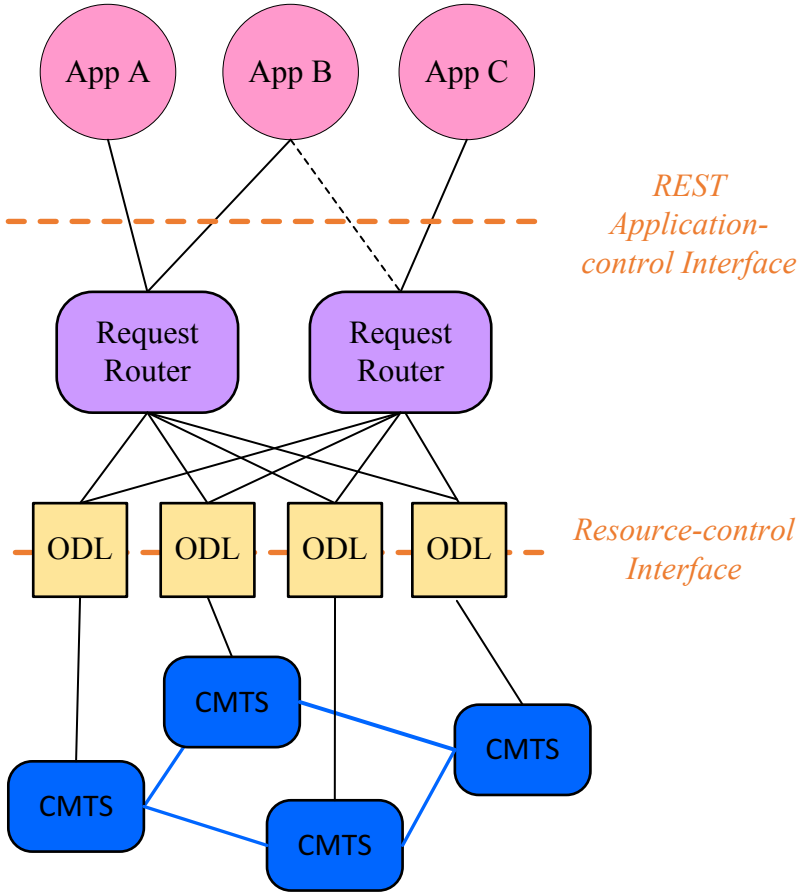


Common Rest NBI allows for direct access or via application. Management activities are done through same API

Common REST SBI based on ODL interface separates core control layer from resource control layer

Practical Architecture

- Application Examples
  - Telephony
  - Congestion Management
  - Acceptable Use Enforcement
- Request Routers in Core
  - Future: decisions/modifications
- ODL at the Edge
  - Independent, swappable (i.e. doesn't have to be ODL)



Routing Core

- Core to provide basic routing of requests now
- In the future, the core will need to possibly do the following:
  - Enhanced, condition-based routing modifications
  - Modification of request contents based on observations/rules (e.g. Time of day/event-based mods)
  - Automatic management activities (not based on requests, self-initiated)
  - Injection of these changes in real time.

Modularity: For Now and The Future

You may have noticed that ODL has been labeled replaceable…

- ODL is maturing, but has issues especially where enhanced features are required for application in DOCSIS network
- In early stages, ODL is not required.
  - As long as the interfaces are the same, we can swap in alternatives for ODL IFF ODL is not maturing sufficiently for production deployment
  - Can always go back to gain access to features IFF ODL matures sufficiently in required areas
- Features/capabilities can be added and removed based on common interface

REST Example

- REST Interface is based on ODL interface
- Consists of URL and JSON attachment defining gate

```
/restconf/config/packetcable:qos/apps/app/<appId>/subscribers/subscriber/<CM_IP>/gates/gate/<gateId>/
```


```
{
    "gate": {
        "gateId": "<gateId>",
        "classifiers": {
            "classifier-container": [
                {
                    "classifier-id": "1",
                    "classifier": {
                        "srcIp": "10.20.0.3",
                        "dstIp": "10.20.0.5",
                        "protocol": "0",
                        "srcPort": "54322",
                        "dstPort": "4322",
                        "tos-byte": "0x01",
                        "tos-mask": "0x00"
                    }
                }
            ]
        },
        "gate-spec": {
            "dscp-tos-overwrite": "0x00",
            "dscp-tos-mask": "0x00",
            "direction": "us"
        },
        "traffic-profile": {
            "flow-spec-profile": {
                "token-bucket-rate": "400",
                "token-bucket-size": "40",
                "peak-data-rate": "400000",
                "minimum-policed-unit": "400",
                "maximum-packet-size": "400",
                "rate": "30000",
                "slack-term": "0"
            }
        }
    }
}
```

Lessons Learned

- **Open source has its challenges**
  - Rapid development? Check. Going where you want? Eh….
  - Acceptable operational practices in one industry may not translate to others.
  - Bad dev decisions: OpenDaylight needs to access the internet on startup
- **Open source needs time to mature**
  - PCMM plugin functional but incomplete for broad application. Be prepared for additional work to bring the parts you need up to speed.
- **Just because it says “Java” doesn’t mean it is good**
  - Something that may be “industry standard” doesn’t mean “industry leading.”
- **Care needs to be taken not to overload terminology**
  - “GateID” turned out to be overloaded in OpenDaylight and the PCMM plugin. Pick terms & stick to them, no ambiguity

“How do you eat an elephant”?

- “One bite at a time”
- Evolutionary approach
  - Do one thing well, move on
  - Don’t boil the ocean
  - Only implement what is needed
  - Leverage other peoples work
  - Remember scale and flexibility
- Keep it Modular

SCTE ISBE CABLE-TEC EXPO '16

SEPTEMBER 26-29 PHILADELPHIA

David S. Early, Ph.D.

david@appliedbroadband.com

SCTE Society of Cable Telecommunications Engineers

ISBE International Society of Broadband Experts

#CableTecExpo

Essential Knowledge for Cable Professionals™